



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

An Architectural Tour of BlueGene/L

D. Dossa

December 2, 2004

NECDC 2004

Livermore, CA, United States

October 4, 2004 through October 7, 2004

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

An Architectural Tour of BlueGene/L

Don Dossa

Center for Applied Scientific Computing, Lawrence Livermore National Lab

L-560 7000 East Ave, Livermore CA 94550

BlueGene/L is the next large supercomputer for the DOE ASC program. BlueGene/L will consist of 65,000 dual processor IBM PowerPC 440 processors, each with attached FPU. Several interconnect networks consisting of a full 3D torus, combining tree, barrier tree, and interrupt tree are used to maximize computing efficiency. The theoretical peak performance of the system is 360 Teraflops/s.

Introduction

BlueGene/L has been designed and built by IBM in close cooperation with the US Department of Energy, with Lawrence Livermore National Lab having a lead technical role. The system, to be delivered in late 2004 through early 2005, will consist of 131,072 IBM PowerPC 440 CPUs, each of which has a dual 64 bit wide floating point processor. The system runs at 700 MHz, providing a theoretical peak of 360 Teraflop/s.

Processor description

The PowerPC 440 core used in BlueGene/L is a 2-way superscalar CPU with 3 execution pipelines. The load/store pipe can have 3 loads outstanding to the memory system. The execution unit permits out of order execution. The ISP includes the usual simple integer and branch instructions. It is a 32 bit wide CPU with 32 general purpose registers. This processor core was designed for embedded applications, so great attention was paid to minimize power consumption with the processor core.

A special, BlueGene/L-unique FPU was attached to each CPU core. Called the Double Hummer FPU, it is a dual, 64-bit wide FPU, containing 32 64-bit registers for each side of the Double Hummer. Both register files can be read from each floating point pipeline. As a result, a floating point instruction can drive either or both FPU pipes on each instruction. The hardware supports SIMD style multiply-add instructions as well as complex arithmetic. By issuing two multiply-add instructions per processor, with 2 processors per node, and 65,536 nodes running at 700 MHz, BlueGene/L yields the theoretical peak of 367 Teraflop/s. As of the November, 2004 Top500 list, a BlueGene/L at just one quarter of its final size was officially the fastest computer in the world.

Torus and Tree Networks

The system connects each node with both a full 3D torus and several tree networks. The torus is used by the MPI library for message passing. The torus is configured as a 32x32x64 torus and is capable of sustaining 175 MiByte/s on each of the 6 nearest neighbor links in bi-directional mode. This yields a 358 GiByte/s bisection bandwidth. MPI uses the torus for sending and receiving messages. Special hardware support allows a message to be broadcast down a given line or plane within the torus. The routing in the torus can be selected between adaptive and deterministic deadlock-free routing. MPI latency on the torus has been measured at 5 microseconds.

Other MPI operations such as broadcast, reduce, and allreduce, use the combining tree network. Any node can be a root of the tree. Packets can be up to 256 bytes long. The combining tree supports in hardware typical MPI collective operations on integers such as sum, min, and max. These operations take place across the entire system in 7 microseconds or less.

MPI Barrier calls are important for synchronizing computations among many MPI processes. As the process count gets into the thousands, typical switch fabric hardware become inefficient for this operation. Overheads of 20-30% have been seen on typical applications running on several thousand processors. BlueGene/L has a dedicated hardware network for the Barrier calls. The Barrier signal can be propagated across the entire compute complex in 5 microseconds.

There is also a dedicated interrupt network in the system that has a similar 5 microsecond latency.

Memory Subsystem

The memory subsystem of a BlueGene/L node is very simple. Each CPU has its own 32 KiB I-stream and D-stream L1 cache. The L2 prefetch and L3 cache are shared between the CPUs on each node. The L3 is 4 MiBytes in size and has a software controllable configuration, although it typically runs as 2 banks of an 8 way set associative cache. The L3 cache supports a 22 GiByte/s bandwidth to the processors. There is also a 16 KiB scratchpad RAM shared by the 2 processors. The DDR controller is a 144 bit wide path to external DDR chips running at 5.5 GiByte/s. While low power consumption was a design goal for the system, low latency to memory was also important. External memory access time is 86 processor clocks, as compared to several hundred processor clocks in modern, commercial CPUs. Since there is no local storage on any of the compute nodes, the memory is mapped once by the kernel at boot/job start time. There is a hard limit of 512 MiBytes of memory per node. Any attempt to access more memory causes a program termination.

IO Subsystem

There is no mass storage of any type in the BlueGene/L system. To access disks, groups of 64 compute nodes are connected through the combining tree with a dedicated IO node. The IO node is identical to a compute node except that each IO node has a Gigabit Ethernet link on it. This link is actually contained inside every compute ASIC as well, but it is only on the IO nodes that this function is connected externally to the ASIC. The Gigabit Ethernet lines are used to connect BlueGene/L to an external disk farm. The ratio of compute nodes to IO nodes is variable; LLNL has chosen a 64:1 ratio. Therefore,

with 64Ki compute nodes, there are an additional 1,024 IO nodes in the system. At LLNL, the BlueGene/L system will be connected to a 900 TeraByte Lustre file system. The external connectivity will provide a usable 30-40 GiByte/s bandwidth to the disk file system for user applications.

System Packaging

The packaging of a node is very simple. There is 1 ASIC designed by IBM which contains everything except the 9 external DRAMs. The ASIC has both 440 cores, the FPU's, the caches, torus, and tree hardware. The ASIC is 11 mm on each side and is fabricated by IBM.

Two nodes are packaged onto 1 small compute card. This card has the 2 ASICs and 18 DRAMs and is approximately 8 inches by 2 inches by 2 inches. Sixteen compute cards are plugged into a node card assembly. The resulting 32 nodes form a 4x4x2 torus. Plugging 16 node cards into a midplane assembly yields a 512 node midplane configured as a 8x8x8 torus. The midplane is the scalable unit of the system, with each midplane delivering nearly 3 Teraflop/s of computation while consuming only 10 KW of power. Two midplanes are placed into each rack. LLNL will have a 64 rack system delivered, with 128 midplanes with a total of 64Ki nodes. This entire system fits into a floor space of the order of a 100 square meters and consumes about 1.3 MW of power.

System Software and Programming Model

The user view of the system is very similar to a cluster of Linux systems. The parallel programming model is MPI in C, C++, and Fortran; OpenMP is not currently supported. There is a light weight kernel running on the compute nodes that implements the most useful 30 Unix system calls. About an equal number of system calls, including all those related to file IO, are function shipped down the combining tree to the IO nodes. The IO nodes are running a full version of Linux and handle all IO requests for the compute nodes. This function shipping is transparent to the user; the standard Unix file open, close, read, write calls are used.

There are 3 supported uses of the second CPU in each node. The simplest, called Heater mode, essentially holds the second CPU in a tight, cache bound loop and this CPU does no work for the user. All user computation work is done on the primary CPU. Communication level software also runs on the primary CPU. The second mode is referred to as Communications Co-Processor mode. This is the intended use of the system. While the user code executes on one CPU, the other CPU is used to control all data handling functions of the torus and tree networks as required by the MPI system calls. This is a preferred mode when the computation and communication loads of the program are in reasonable balance. The final mode is Virtual Node Mode. This permits the user to use both CPUs on each node with each CPU running one MPI process. This is not a shared memory model; the memory is split between the CPUs. A small section is maintained by the kernel to optimize message passing between the MPI processes on the node. This is the most efficient mode when the program is heavily computation bound.

Because the compute nodes are running a light weight kernel, there are no other processes running on the compute nodes. This simplifies the design of the kernel and guarantees that there is almost no system-generated timing noise. In general, applications will for the most part run in lock-step since there is one system clock and system booting and job initiation is started synchronously by a large, external, front-end node computer.

Acknowledgements

Many people from LLNL have worked on the system definition and requirements for BlueGene/L. A partial list is Mark Seager, Lynn Kissel, Kim Yates, Andy Yoo, Bronis de Supinski, Moe Jette, Dong Ahn and Ryan Braby.

This work was performed under the auspices of the US Department of Energy by the University of California Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.

References

The BlueGene/L Team, IBM and Lawrence Livermore National Lab, “An overview of the BlueGene/L Supercomputer”, SC02, Nov, 2002.

Almasi, G., Archer, C., Castanos, J., et al, “ Design and Implementation of message passing services for the BlueGene/L Supercomputer,” Accepted for publication, IBM Journal of Research and Development.

Adiga, N., Blumrich, M., Chen, D., Coteus, P., Gara, A., Giampapa, E., Heidelberger, P. “BlueGene/L torus interconnection network”, Accepted for publication, IBM Journal of Research and Development.

Davis, K., Hoisie, A., Johnson, G., Kerbyson, D., Lang, M., Pakin, S., Petrini, F., “A Performance and Scalability Analysis of the BlueGene/L Architecture”, presented SC04, Pittsburgh, Nov 2004.